UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/630,635	07/29/2003	Jeffrey Jay Rooney	MTIPAT.002C1C1	9052

20995 7590 10/02/2007

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
PARK, ILWOO	

ART UNIT	PAPER NUMBER
2182	

NOTIFICATION DATE	DELIVERY MODE
10/02/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

jcartee@kmob.com
eOAPilot@kmob.com

Office Action Summary	Application No. 10/630,635	Applicant(s) ROONEY, JEFFREY JAY	
	Examiner Ilwoo Park	Art Unit 2182	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 July 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-7,9-12,14-16,23-29,31-35,37 and 39-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-7,9-12,14-16,23-29,31-35,37 and 39-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 7/9/07 has been entered.

2. Claims 2, 10, 23, 31, 37, and 44 are amended and claim 49 is added in response to the last office action. Solari, Amini et al, and Rabe et al were cited in the last office action. Claims 2-7, 9-12, 14-16, 23-29, 31-35, 37 and 39-49 are presented for examination

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 2-7, 9-12, 14-16, 23-29, 31, 33-35, 37, 39-42, and 44-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Solari [US 5,333,276] in view of Amini et al. [US 5,644,729] and further in view of Carlson et al. [US 5,692,200].

As to claim 2, Solari teaches a method for providing data transfers between a processor [processor 122 in fig. 1b] and a component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b], the method comprising:

handling requests originating from a processor by buffering [col. 2, lines 57-59] a first address with a first address buffer [e.g., register 1 207 in fig. 2a] and a second address with a second address buffer [e.g., register 2 208 in fig. 2a], the first and second address buffers being in communication with a processor and a component, wherein the processor operates at a different speed [col. 5, lines 5-12] than the component;

buffering [col. 2, lines 57-59] a first data value with a first data buffer [e.g., register 1 227 in fig. 2b] and buffering a second data value with a second buffer [e.g., register 2 228 in fig. 2b], the first and second buffers [col. 2, lines 55-57] being in communication with the processor and the component;

reading ["by checking the flag field 302" in col. 9, lines 39-40] status information from the first address buffer to determine a priority status [such as PRIORITY READ, PRIORITY WRITE in fig. 3b] of the first data value;

reading ["by checking the flag field 302" in col. 9, lines 39-40] status information from the second address buffer to determine a priority status [such as PRIORITY READ, PRIORITY WRITE in fig. 3b] of the second data value;

controlling [col. 7, lines 35-52] the first address buffer and the first data buffer as a matched pair such that the first address held in the first address buffer corresponds to the first data value held in the first buffer;

controlling [col. 7, lines 35-52] the second address buffer and the second data buffer as a matched pair such that the second address held in the second address buffer corresponds to the second data value held in the second buffer; and

controlling [col. 11, lines 11-28; col. 13, lines 48-61] the order of bi-directional data flow such that data flows between the processor and the component, wherein the controlling the order of the bi-directional data flow is based on the priority status of the first and second data values.

Solari does not expressly disclose the data buffers used for data transfers between two buses are bi-directional buffers. Further, Solari does not disclose controlling the order of bi-directional data flow through the bi-directional data buffers. Amini et al teach bi-directional [col. 6, lines 30-38] data buffers buffering data values for providing data transfers between a processor and a component. Amini et al further teach controlling [col. 9, lines 32-63] the order of bi-directional data flow through the bi-directional data buffers based on a priority status of data values. At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the cited references in order to increase flexibility of bi-directional data transfers by allowing simultaneous operation between two buses as taught by Amini et al [col. 6, lines 30-38]. However, the combination of Solari and Amini et al does not expressly teach routing requests originating from a component to the processor through a target controller. Carlson et al teach a method for providing data transfers between a processor [SYSTEM PROCESSOR 104 in fig. 1] and a component ["PCI bus master" in col. 4, lines 55-61], the method comprising routing requests ["interrupt signal" in col. 5,

lines 49-56] originating from a component to the processor through a target controller [INTERRUPT CONTROL 264 in figs. 3 and 4] and handling requests originating from the processor by other controller. Therefore, it would have been obvious to one of the ordinary skill in the art at the time the invention was made to include a target controller routing requests originating from a component to a processor in order to increase reliability of the data transfer from the processor to the component.

5. As to claim 3, Solari teaches the first and second buffers are in communication with the processor via a bus [fig. 1a].

6. As to claim 4, Solari teaches the first and second buffers are in communication with a bus master controller and a bus slave controller [address FIFO input/output controls in fig. 2a].

7. As to claim 5, Solari teaches the first address buffer further comprises status bits [figs. 3a-c].

8. As to claim 6, Solari teaches the status bits relate to the type of request being made by the processor [figs. 3a-3c].

9. As to claim 7, Solari teaches said controlling said first address buffer and the first data buffer as matched pair is performed with pointers [col. 9, lines 15-23].

10. As to claim 9, Solari teaches said act of controlling bi-directional data flow is performed with at least one input data arbiter [col. 9, lines 12-23].

11. As to claim 10, Solari teaches a method for controlling data transfers between a processor [processor 122 in fig. 1b] and a component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b], the method comprising:

buffering [col. 2, lines 57-59] with a plurality of address buffers address requests originating from a processor to a component, wherein the processor operates at a different speed [col. 5, lines 5-12] than the component;

storing status information [e.g., FLAG 302 of address register including PRIORITY READ, PRIORITY WRITE in figs. 3a and 3b] in each of the plurality of address buffers, the status information determining the priority status of data transfers associated with the address requests;

bi-directionally buffering [col. 2, lines 55-59] with a plurality of data buffers data transfers between the processor and the component, wherein said data transfers can be performed out of a previously defined order [col. 8, lines 4-5; col. 11, lines 11-28; col. 13, lines 48-61] based on ["by checking the flag field 302" in col. 9, lines 39-40] the priority status of each of the data transfers; and

controlling [col. 7, lines 35-56] said buffering address requests and said bi-directionally buffering such that each of the buffered data transfers relates to an address held in one of the plurality of address buffers.

Solari does not expressly disclose the data buffers used for data transfers between two buses are bi-directional buffers. Amini et al teach bi-directional [col. 6, lines 30-38] data buffers buffering data values for providing data transfers between a processor and a component. At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the cited references in order to increase flexibility of bi-directional data transfers by allowing simultaneous operation between two buses as taught by Amini et al [col. 6, lines 30-38]. However, the combination of Solari

and Amini et al does not expressly teach routing requests originating from a component to the processor through a target controller. Carlson et al teach a method for providing data transfers between a processor [SYSTEM PROCESSOR 104 in fig. 1] and a component ["PCI bus master" in col. 4, lines 55-61], the method comprising routing requests ["interrupt signal" in col. 5, lines 49-56] originating from a component to the processor through a target controller [INTERRUPT CONTROL 264 in figs. 3 and 4] and handling requests originating from the processor by other controller. Therefore, it would have been obvious to one of the ordinary skill in the art at the time the invention was made to include a target controller routing requests originating from a component to a processor in order to increase reliability of the data transfer from the processor to the component.

12. As to claim 11, Solari teaches indicating which of the plurality of data buffers is available to accept new data [col. 10, lines 47-53].

13. As to claim 12, Solari teaches said act of indicating is performed with reference pointers [col. 9, lines 12-23].

14. As to claim 14, Solari teaches said act of buffering address requests includes the use of an input arbiter and an output arbiter [col. 9, lines 12-23].

15. As to claim 15, Solari teaches said act of bi-directional buffering is performed with an input arbiter and an output arbiter [col. 9, lines 12-23].

16. As to claim 16, Solari teaches the plurality of address buffers comprises at least three address buffers and wherein the plurality of data buffers comprises at least three data buffers [figs. 2a-2b].

17. As to claim 23, Solari teaches a method of transferring addresses and data through a buffer [col. 2, lines 55-57], the method comprising:

storing [col. 2, lines 57-59] a first address in a first buffer in communication with a first component and a second component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b], the first buffer comprising status bits [e.g., FLAG 302 of address register including PRIORITY READ, PRIORITY WRITE in figs. 3a and 3b];

storing [col. 7, lines 35-52] first data in a second buffer matched with said first buffer so that the first address storied in the first buffer is related to the first data stored in the second buffer;

storing [col. 2, lines 57-59] a second address in a third buffer in communication with a first component and a second component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b], the third buffer comprising status bits [e.g., FLAG 302 of address register including PRIORITY READ, PRIORITY WRITE in figs. 3a and 3b];

storing [col. 7, lines 35-52] second data in a fourth buffer matched with said third buffer so that the second address storied in the third buffer is related to the second data stored in the fourth buffer;

reading ["by checking the flag field 302" in col. 9, lines 39-40] the status bits of the first buffer to determine [col. 11, lines 11-28; col. 13, lines 48-61] a first priority value of the first data;

reading ["by checking the flag field 302" in col. 9, lines 39-40] the status bits of the second buffer to determine [col. 11, lines 11-28; col. 13, lines 48-61] determining a second priority data value on the second data; and

controlling [col. 11, lines 11-28; col. 13, lines 48-61] the order of data flow of the first data and the second data based at least in part on said first and second priority values.

Solari does not expressly disclose the second and fourth data buffers used for data transfers between two buses are bi-directional buffers. Further, Solari does not disclose controlling the order of bi-directional data flow through the bi-directional data buffers. Amini et al teach bi-directional [col. 6, lines 30-38] data buffers buffering data values for providing data transfers between a processor and a component. Amini et al further teach controlling [col. 9, lines 32-63] the order of bi-directional data flow through the bi-directional data buffers based on priority data values. At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the cited references in order to increase flexibility of bi-directional data transfers by allowing simultaneous operation between two buses as taught by Amini et al [col. 6, lines 30-38]. However, the combination of Solari and Amini et al does not expressly teach routing requests originating from a component to the processor through a target controller. Carlson et al teach a method for providing data transfers between a processor [SYSTEM PROCESSOR 104 in fig. 1] and a component ["PCI bus master" in col. 4, lines 55-61], the method comprising routing requests ["interrupt signal" in col. 5, lines 49-56] originating from a component to the processor through a target controller [INTERRUPT CONTROL 264 in figs. 3 and 4] and handling requests originating from the processor by other controller. Therefore, it would have been obvious to one of the ordinary skill in the art at the time the invention was made to include a target controller

routing requests originating from a component to a processor in order to increase reliability of the data transfer from the processor to the component.

18. As to claim 24, Solari teaches the status bits comprise transfer type bits indicative of the status of an address transfer from the first component to the first buffer [figs. 3a-3c].

19. As to claim 25, Solari teaches the status bits comprise transfer type bits indicative of the status of a data transfer from the first component to the first buffer [figs. 3a-3c].

20. As to claim 26, Solari teaches the first component comprises a memory [fig. 1b].

21. As to claim 27, Solari teaches the first component comprises a processor [fig. 1b].

22. As to claim 28, Solari teaches the first buffer is in communication with the processor via a bus [fig. 1b].

23. As to claim 29, Solari teaches the first buffer is in communication with the processor via a bus master controller and a bus slave controller [address FIFO input/output controls in fig. 2a].

24. As to claim 31, Solari teaches a method of transferring data between a between a processor [processor 122 in fig. 1b] and a component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b] utilizing a plurality of address buffers and a plurality of data buffers [figs. 2a-2b], the method comprising:

when a request originates from the processor, receiving [col. 13, lines 48-68] a data request including an associated address from a processor;

determining [col. 10, lines 47-53] whether at least one of a plurality of address buffers and an associated [col. 7, lines 35-52] data buffer are available;

storing [col. 2, lines 57-59] the associated address in the at least one address buffer;

storing status information [e.g., FLAG 302 of address register including PRIORITY READ, PRIORITY WRITE in figs. 3a and 3b] indicative of a priority of the data request in the at least one address buffer;

receiving [fig. 8a] data identified by the associated address from the component with the data buffer

ordering [fig. 8a; col. 13, lines 48-68], based on the priority of the data request, the transmission of the data from the data buffer to the processor.

Solari does not expressly disclose the data buffers used for data transfers between two buses are bi-directional buffers. Amini et al teach bi-directional [col. 6, lines 30-38] data buffer buffering data values for providing data transfers between a processor and a component. Amini et al further teach ordering [col. 9, lines 32-63] the transmission of the data from the bi-directional data buffer based on a priority of the data request. At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the cited references in order to increase flexibility of bi-directional data transfers by allowing simultaneous operation between two buses as taught by Amini et al [col. 6, lines 30-38]. However, the combination of Solari and Amini et al does not expressly teach routing request through a target controller when the request originate from the component. Carlson et al teach a method for providing data

transfers between a processor [SYSTEM PROCESSOR 104 in fig. 1] and a component ["PCI bus master" in col. 4, lines 55-61], the method comprising routing requests ["interrupt signal" in col. 5, lines 49-56] originating from a component to the processor through a target controller [INTERRUPT CONTROL 264 in figs. 3 and 4] and handling requests originating from the processor by other controller. Therefore, it would have been obvious to one of the ordinary skill in the art at the time the invention was made to include a target controller routing requests originating from a component to a processor in order to increase reliability of the data transfer from the processor to the component.

25. As to claim 33, Solari teaches the at least one address buffer and the data buffer are in communication with the processor via the bus [fig. 1b].

26. As to claim 34, Solari teaches the at least one address buffer and the data buffer are in communication with the bus via a bus master controller and a bus slave controller [address FIFO input/output controls in fig. 2a].

27. As to claim 35, Solari teaches the data buffer and the at least one address buffer are associated with each other through the use of pointers [col. 9, lines 12-23].

28. As to claims 37 and 49, Solari teaches an apparatus for controlling data transfers between a processor [processor 122 in fig. 1b] and a component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b], the apparatus comprising:

means for buffering [col. 2, lines 55-59] address requests from a processor to a component;

means for buffering [col. 2, lines 55-59] data transfers between the processor and the component;

means for storing status information [e.g., FLAG 302 of address register including PRIORITY READ, PRIORITY WRITE in figs. 3a and 3b] indicative of a priority status of buffered data transfer; and

means for controlling [col. 7, lines 35-52] the means for buffering and the means for buffering so that each of the buffered data transfers relates to an address held in the means for buffering, wherein the means for controlling further coordinates [col. 8, lines 58-61; col. 11, lines 11-28; col. 13, lines 48-61] said data transfers based at least on a priority status of each buffered data transfer.

Solari does not expressly disclose means for buffering data transfers the data buffers between two buses are bi-directional. Amini et al teach means for bi-directionally [col. 6, lines 30-38] buffering data transfers between a processor and a component. At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the cited references in order to increase flexibility of bi-directional data transfers by allowing simultaneous operation between two buses as taught by Amini et al [col. 6, lines 30-38]. However, the combination of Solari and Amini et al does not expressly teach routing request from the component from the processor. Carlson et al teach a method for providing data transfers between a processor [SYSTEM PROCESSOR 104 in fig. 1] and a component ["PCI bus master" in col. 4, lines 55-61], the method comprising routing requests ["interrupt signal" in col. 5, lines 49-56] originating from a component to the processor through a target controller [INTERRUPT CONTROL 264 in figs. 3 and 4] and handling requests originating from the processor by other controller. Therefore, it would have been obvious to one of the ordinary skill in the

art at the time the invention was made to include a target controller routing requests originating from a component to a processor in order to increase reliability of the data transfer from the processor to the component.

29. As to claim 39, Solari teaches means for buffering includes a plurality of address buffers [fig. 2a].

30. As to claim 40, Solari teaches means for buffering includes a plurality of data buffers [fig. 2b].

31. As to claim 41, Solari teaches means for buffering includes an input arbiter and an output arbiter [col. 9, lines 12-23].

32. As to claim 42, Solari teaches means for buffering includes an input arbiter and an output arbiter [col. 9, lines 12-23].

33. As to claim 44, Solari teaches a buffer allocation system for managing data flow between components [e.g., ref. Nos. 122, 132-135, 142-144 in fig. 1b] of a computer, the system comprising:

an address buffer module configured to handle address requests between a first component [processor 122 in fig. 1b] and a second component [e.g., ref. Nos. 132-135, 142-144 in fig. 1b] for requests originating from the first component, the address buffer module comprising: a plurality of address buffers [e.g., registers 207-209 in fig. 2a] each in communication with the first and second components, each address buffer comprising status information [e.g., FLAG 302 of address register including PRIORITY READ, PRIORITY WRITE in figs. 3a and 3b] indicative of a priority status;

an input address arbiter [col. 9, lines 12-23] configured to direct the address requests to the plurality of address buffers, and an output address arbiter [col. 9, lines 12-23] configured to send the address requests from the plurality of address buffers to the second component; and

a data buffer module configured to control an order [col. 8, lines 58-61; col. 11, lines 11-28; col. 13, lines 48-61] of bi-directional flow of data therethrough based on the priority status of the data, the data buffer module comprising: a plurality of data buffers [e.g., registers 227-229 in fig. 2b] each in communication with the first and the second components, an input data arbiter [col. 9, lines 12-23] configured to direct data to the plurality of data buffers, and an output data arbiter [col. 9, lines 12-23] configured to direct data output from the plurality of data buffers.

Solari does not expressly disclose the data buffers used for data transfers between two buses are bi-directional buffers. Further, Solari does not disclose controlling the order of bi-directional data flow through the bi-directional data buffers. Amini et al teach bi-directional [col. 6, lines 30-38] data buffers buffering data values for providing data transfers between a processor and a component. Amini et al further teach controlling [col. 9, lines 32-63] the order of bi-directional data flow through the bi-directional data buffers based on a priority status of data values. At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the cited references in order to increase flexibility of bi-directional data transfers by allowing simultaneous operation between two buses as taught by Amini et al [col. 6, lines 30-38]. However, the combination of Solari and Amini et al does not

expressly teach routing requests originating from the second component through a target controller. Carlson et al teach a method for providing data transfers between a first component [SYSTEM PROCESSOR 104 in fig. 1] and a second component ["PCI bus master" in col. 4, lines 55-61], the method comprising routing requests ["interrupt signal" in col. 5, lines 49-56] originating from the second component to the first component through a target controller [INTERRUPT CONTROL 264 in figs. 3 and 4] and handling requests originating from the first component by other controller. Therefore, it would have been obvious to one of the ordinary skill in the art at the time the invention was made to include a target controller routing requests originating from a component to a processor in order to increase reliability of the data transfer from the processor to the component.

34. As to claim 45, Solari teaches the plurality of address buffers comprising a first address buffer and the plurality of data buffers comprising a first data buffer, wherein an address held in the first address buffer corresponds only to a data value held in the first data buffer [col. 7, lines 35-52].

35. As to claim 46, Solari teaches the plurality of address buffers comprising a second address buffer and the plurality of data buffers comprising a second data buffer, wherein an address held in the second address buffer corresponds only to a data value held in the second data buffer [col. 7, lines 35-52].

36. As to claim 47, Solari teaches a plurality of multiplexers coupled to the output address arbiter and the plurality of address buffers [see fig. 2a].

37. As to claim 48, Solari teaches a plurality of multiplexers coupled to the input data arbiter and the plurality of data buffers [see fig. 2b].

38. Claims 32 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Solari [US 5,333,276] in view of Amini et al. [US 5,644,729] and further in view of Carlson et al [US 5,692,200] as applied to claims 31 and 23 above, and further in view of Rabe et al. [US 5,664,122].

As to claims 32 and 43, Solari, Amini et al, and Carlson et al teach controlling the separate address buffer and the separate bi-directional data buffer as a matched pair. However, Solari, Amini et al, and Carlson et al do not explicitly disclose the buffer control allowing data to be read from the separate bi-directional data buffer while an address is written to the separate address buffer. Rabe et al teach a method for providing data transfers between a processor and a component controlling a separate address buffer and a separate data buffer and the buffer control allowing data to be read from a first data buffer while [col. 8, lines 56-64; col. 9, lines 43-51] an address is written to a first address buffer. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the buffer control allowing data to be read from a first data buffer while an address is written to a first address buffer in order to expedite the immediate priority data transfer control between a processor and a component of Solari, Amini et al, and Carlson et al.

Conclusion

39. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ilwoo Park whose telephone number is (571) 272-4155. The examiner can normally be reached on Monday through Friday from 9:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Huynh can be reached on (571) 272-4147. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ILWOO PARK
PRIMARY EXAMINER

Ilwoo Park

Ilwoo Park
September 14, 2007